Homework 8 - Sistema de Urnas

Claudia Lizeth Hernández Ramírez

20 de octubre de 2021

Resumen

Conforme se aumentan las iteraciones, el porcentaje de filtración disminuye considerablemente. El mejor porcentaje de filtración resulta en la primer iteración.

1. Introducción

Simulando un sistema de filtración de aguas residuales, se formas cúmulos de partículas las cuales pueden fragmentarse o unirse entre ellas. Se determina un número fijo de partículas `n`, en este caso `100000` y se estarán variando los cúmulos `k` entre `100`, `200` y `400`, el tamaño del `filtro` será igual a `c`, es decir que tamaño `crítico`.

2. Desarrollo

Se trabajo con el código base que se explicó en clase [2]. En un inicio se agregaron dos ciclos `FOR`, el primero para variar los cúmulos, denomiados en el código como `grupo`; el segundo ciclo `FOR` es referente a las réplicas que realizará el programa, en esta ocasión fueron `50` réplicas.

Listing 1: Segmento de código ciclos `FOR` para cúmulos y repeticiones.

```
grupo <- c(100, 200, 400)
replica = 1:50
n <- 100000
datos = data.frame()

for (k in grupo){
  for (reply in replica){
    originales <- rnorm(k)
    cumulos <- originales - min(originales) + 1
    cumulos <- round(n * cumulos / sum(cumulos))
```

Posteriormente, basándome en los ejemplos vistos en clase [3], se generó un código para sacar el porcentaje de filtración exitoso.

.

Listing 2: Segmento de código para filtrado exitoso.

```
filtrados = freq[freq$tam >= c,]
      filtrados$cont = filtrados$tam * filtrados$num
      f = sum(filtrados$cont) # particulas removidas
      porcentaje = (f/n) * 100 # porcentaje exitosamente filtrado
      resultado = c(k, reply, paso, porcentaje)
      datos = rbind(datos, resultado)
      names(datos) = c("k", "Replica", "Iteracion", "Filtrado")
```

Todo esto dentro de los ciclos `FOR` que se generaron en un inicio.

Después apliqué pruebas de normalidad a mi datos.

Listing 3: Segmento de código para prueba de normalidad Shapiro-Wilk.

```
#Estadistica prueba de normalidad -
      #con p menor a 0.05 se rechaza hipotesis nula H0
      #H0: las medias son iguales en todos los grupos.
      #H1: existen diferencias significativas entre las medias de los grupos.
tapply(datos$Filtrado, datos$k, shapiro.test)
```

De esta prueba se obtuvieron los datos expresados en el cuadro 1 (En la literatura podemos encontrar que diversos autores establecen que en pruebas de normalidad para aceptar la hipótesis nula el valor de `P` debe ser mayor al 0.05, es decir mayor al 5 %).

Cuadro 1: Resultados obtenidos de prueba de normalidad de Shapiro.

Cúmulo	W value	P value	¿Se acepta H0?
100	0.41416	$2{,}2 \times 10^{-16}$	no
200	0.83836	$2{,}2 \times 10^{-16}$	no
400	0.99943	0.6802	sí

Con la información obtenida en las pruebas de normalidad, podemos seguir con la prueba estadística. En esta ocasión recurriremos a la prueba de `Kruskal-Wallis` [1].

Listing 4: Segmento de código para prueba de Kruskal-Wallis

```
#PRUEBA ESTADISTICA
datos%>%
 group_by(k) %>%
  summarise(
    cantidad_de_participantes = n(),
    promedio = mean(Filtrado, na.rm = TRUE),
    desviacion_estandar = sd(Filtrado, na.rm = TRUE),
    varianza = sd(Filtrado, na.rm = TRUE)^2,
    mediana = median(Filtrado, na.rm = TRUE),
    rango_intercuartil = IQR(Filtrado, na.rm = TRUE)
    )

kruskal.test(Filtrado ~ k, data = datos)
pairwise.wilcox.test(datos$Filtrado, datos$k)
```

De donde se obtuvieron los resultados expresados en el cuadro 2, 3 y 4.

Cuadro 2: Resultados obtenidos de prueba Kruskal-Wallis.

Chi cuadrada	DF	P
6600.4	2	$2{,}2 \times 10^{-16}$

Cuadro 3: Diferencias entre grupos. Kruskal-Wallis.

	100	200
200	2×10^{-16}	
400	2×10^{-16}	2×10^{-16}

Cuadro 4: Información individual de los datos.

Probabilidad	Qty. Participantes	promedio	Desv. Std.	Varianza	Mediana	Rango Intercuartil
100	2500	2.26	6.20	38.4	0	1.01
200	2500	27.6	3.36	11.3	27.0	3.07
400	2500	49.9	1.38	1.90	49.9	1.86

La información presentada anteriormente puede visualizarse más facilmente en las gráficas 1, 2 y 3.

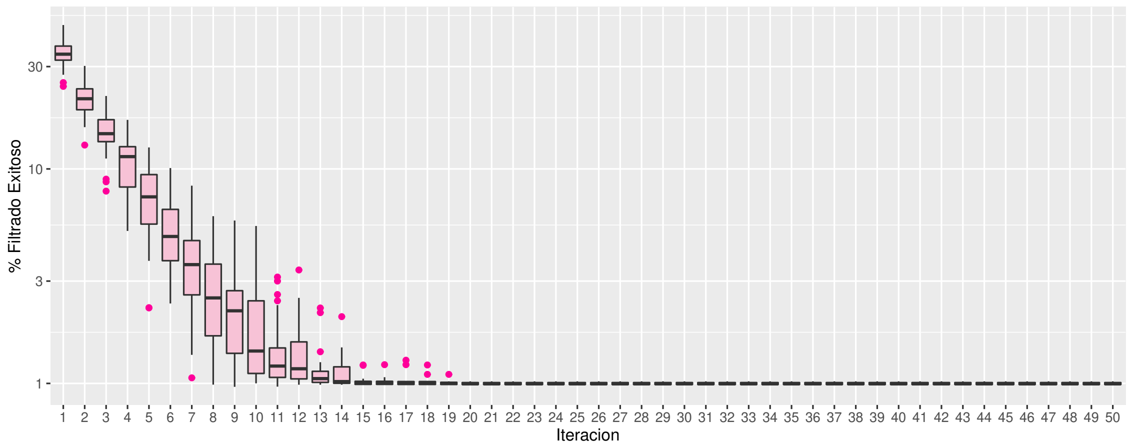

Figura 1: k = 100.

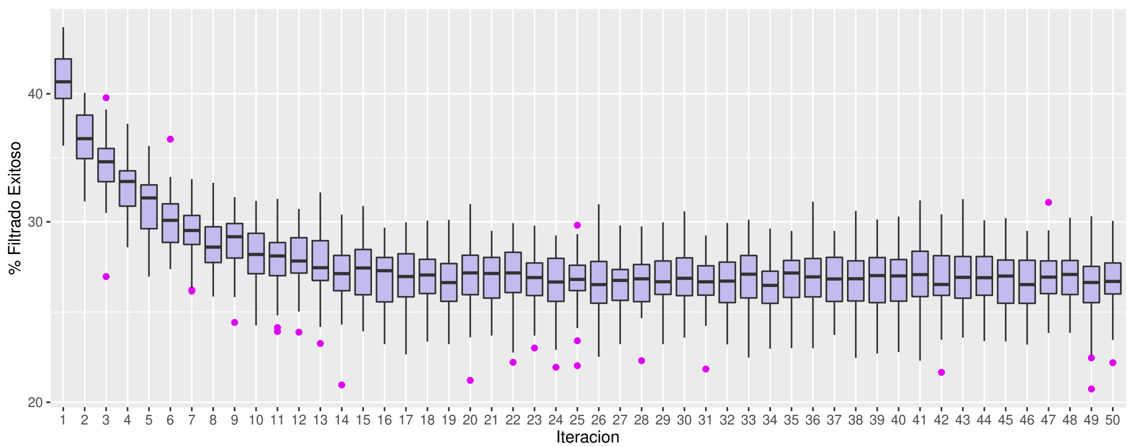

Figura 2: k = 200.

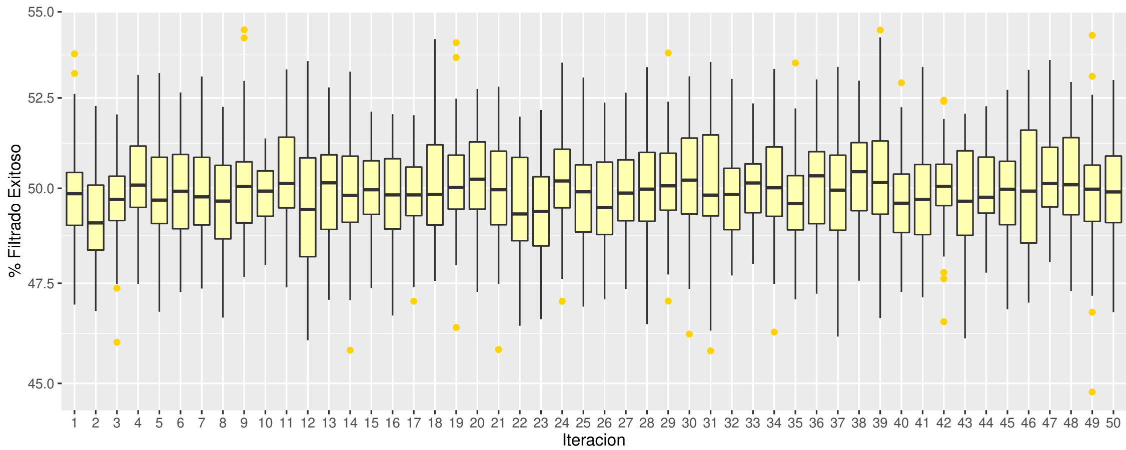

Figura 3: k = 400.

Considerando que

-Hipótesis nula: hay diferencias en el porcentaje de filtración exitoso y las iteraciones.

-Hipótesis alternativa: no hay diferencias en el porcentaje de filtración exitoso y las iteraciones.

-Con un nivel de significancia igual a 0.05

Contamos con evidencia estadística suficiente para rechazar la `hipótesis nula`.

3. Conclusión

Con base en las imágenes anteriores, podemos deducir que el porcentaje de filtración irá disminuyendo conforme las iteraciones aumentan, sin embargo ocurre algo excepcional en el grupo de cúmulos `k=400` ya que no se observa la misma tendencia que en los primeros dos grupos, `k=100` y `k=200`. Ignorando dicho grupo, pareciera que el mejor resultado de filtración resulta de la primera iteración.

4. Reto 2 - Variar tamaño crítico para validar cantidad de partículas que quedan atrapadas en el filtro.

Se modificó el tamaño de `c` que como se mencionó con anterioridad representa el tamaño crítico. En esta ocasión no tomará el valor de la media sino, el valor del cuartil.

Aplicando el mismo código que, en la tarea base, se obtienen las imágenes 4, 5 y 6.

Listing 5: Segmento de código para modificar valor del tamaño crítico.

```
assert(length(cumulos[cumulos == 0]) == 0) # que no haya vacios
    assert(sum(cumulos) == n)
    c <- median(cumulos)/2 # tamanio critico de cumulos
    d <- sd(cumulos) / 4 # factor arbitrario para suavizar la curva
    rotura <- function(x) {
      return (1 / (1 + exp((c - x) / d)))
    }
```

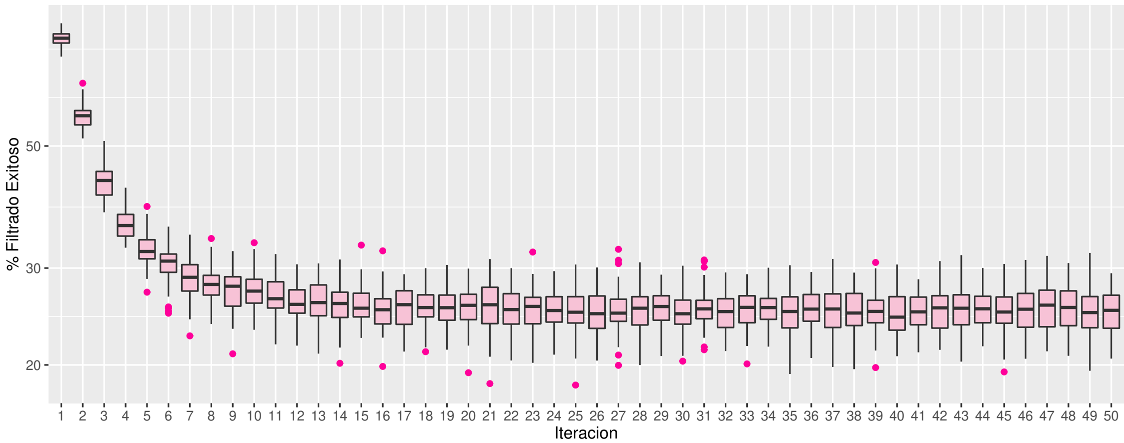

Figura 4: k = 100 R2.

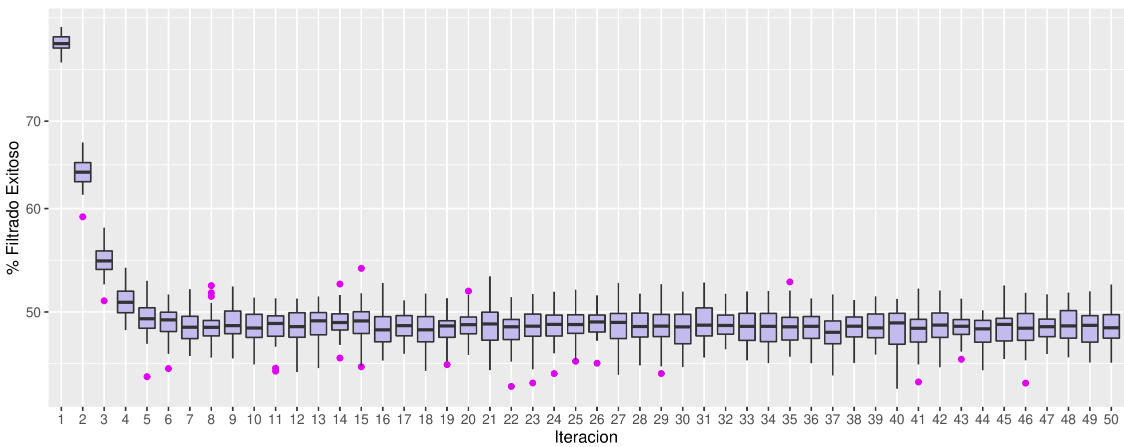

Figura 5: k = 200 R2.

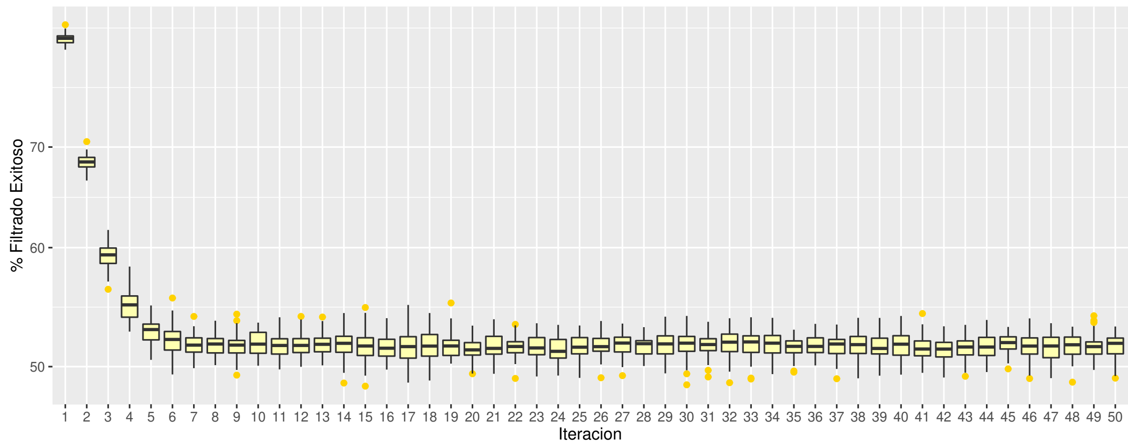

Figura 6: k = 400 R2.

5. Conclusión Reto

Comparando las imágenes obtenidas después de modificar el tamaño crítico, podemos observar una notable mejoría en el porcentaje de filtración. Por mencionar algunas, si comparamos las gráficas 1 y 4, se observa que en la primer iteración se alcanzan valores alrededor de 30 % y 65 % respectivamente[4]. De igual forma se observa con mayor claridad que para los tres grupos de `k` el momento en que maximiza el porcentaje de filtración resulta en la primer iteración.

Referencias

[1] Jose Antonio. Codigo prueba estadistica, 2020. URL `https://www.youtube.com/watch?v=WEjudFpbCcE&list=PLVcE9e4AaRiXSnHyCPUONsGcpakAx5f25&index=5&t=617s`.

[2] Elisa Schaeffer. Codigo clase, 2021. URL `https://github.com/satuelisa/Simulation/blob/master/UrnModel/aggrFrag.R`.

[3] Elisa Schaeffer. Codigo filtrados, 2021. URL `https://github.com/satuelisa/Simulation/blob/master/UrnModel/filtrado.R`.

[4] Wikia.org. Simbolo porcentaje, 2021. URL `https://latex.wikia.org/wiki/Percent_sign_(LaTeX_symbol)`.